Available online at www.sciencedirect.com
SCIENCE DIRECT
Analytica Chimica Acta 566 (2006) 81–98
ANALYTICA CHIMICA ACTA
www.elsevier.com/locate/aca

Properties of a genetic algorithm extended by a random self-learning operator and asymmetric mutations: A convergence study for a task of powder-pattern indexing

Wojciech Paszkowicz *

Institute of Physics, Polish Academy of Sciences, Al. Lotników 32/46, PL-02-668 Warsaw, Poland

Received 22 July 2005; received in revised form 1 February 2006; accepted 20 February 2006

Abstract

Genetic algorithms represent a powerful global-optimisation tool applicable in solving tasks of high complexity in science, technology, medicine, communication, etc. The usual genetic-algorithm calculation scheme is extended here by introduction of a quadratic self-learning operator, which performs a partial local search for randomly selected representatives of the population. This operator is aimed as a minor deterministic contribution to the (stochastic) genetic search. The population representing the trial solutions is split into two equal subpopulations allowed to exhibit different mutation rates (so called asymmetric mutation). The convergence is studied in detail exploiting a crystallographic-test example of indexing of powder diffraction data of orthorhombic lithium copper oxide, varying such parameters as mutation rates and the learning rate. It is shown through the averaged (over the subpopulation) fitness behaviour, how the genetic diversity in the population depends on the mutation rate of the given subpopulation. Conditions and algorithm parameter values favourable for convergence in the framework of proposed approach are discussed using the results for the mentioned example. Further data are studied with a somewhat modified algorithm using periodically varying mutation rates and a problem-specific operator. The chance of finding the global optimum and the convergence speed are observed to be strongly influenced by the effective mutation level and on the self-learning level. The optimal values of these two parameters are about 6 and 5%, respectively. The periodic changes of mutation rate are found to improve the explorative abilities of the algorithm. The results of the study confirm that the applied methodology leads to improvement of the classical genetic algorithm and, therefore, it is expected to be helpful in constructing of algorithms permitting to solve similar tasks of higher complexity.

Keywords: Diffraction pattern; Genetic algorithm; Indexing; Self-learning; Male driven evolution; Mutation; Operator; Premature convergence

1. Introduction

Global search and optimisation methods aim to find (or to approach) a global minimum or at least a 'satisfactory' solution in a large space characterised by a complex shape and many extremes. They are used for solving tasks, which cannot be treated neither in an analytical way nor by using 'hill-climbing' search routines. One of known valuable global-optimisation methods is provided by genetic algorithms (GAs) [1,2] forming a subset of broader classes of global-optimisation strategies called population-based methods and evolutionary algorithms. The application of GAs for optimisation has been initiated by De Jong [3]. GAs are helpful in solving many-parameter optimisation tasks in various domains of science, technology, medicine, etc., where the local 'hill-climbing' techniques (requiring a starting point close to the global solution) are useless.

The concept of GAs follows the old idea of minimizing human efforts in solving difficult scientific and technical problems by learning from nature (for discussion of this concept, see refs. [1,2,4]). The genetic computation proceeds in the space of variables. It mimics the evolution of living organisms represented by points in this space (individuals). In the beginning, initial population of individuals is generated. Next generations are successively created using simplified principles of (Dar-

Abbreviations: DPF, dynamic penalty function; GA, genetic algorithm; HGA, hybrid genetic algorithm; LCO, lithium copper oxide; LM, local minimum; MGA, mixed genetic algorithm; OFN, objective function; RGA, reduced genetic algorithm; SL, self learning

* Tel.: +48 228436601 3301/034; fax: +48 228430926.
E-mail address: paszk@ifpan.edu.pl.

0003-2670/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.aca.2006.02.055

winian) evolution. The calculation is terminated according to a stop condition such as predetermined fixed number of generations or some convergence criterion. The variables are organised in strings composed of real or integer numbers, vectors, matrices, logical variables . . .; frequently, the variables are binary coded. The basic genetic operators used in formation of each new population include selection, crossover (both may be classified as co-operative operators) and mutation (classified as self-adaptive one). The search in population-based methods is most frequently elitist (i.e. the principle of 'survival of 35 the fittest' prevents the already found "optimal" solution from extinction).

For the given optimisation task, a fitness function describing the population-member quality must be defined together with the variable ranges and other problem-specific constraints. The GA minimises (maximises) an objective function (OFN), which is either identical to fitness or is defined as linearly/nonlinearly scaled fitness. The selection operator ascertains a greater chance to pass the "good" genes to next generations for better-adapted individuals (solutions). This operator determines which individuals are chosen for crossover, whereas the crossover operator determines how the bits are exchanged between the individuals. Selection can be done on the basis of the OFN values (e.g., using the roulette-wheel method, on the ranking, or using the tournament selection). A part of individuals are submitted to crossover (with predefined crossover probability), the remaining ones are simply transferred to the next generation. A mutation operator typically consists of flipping a single bit with a small probability (mutation rate, m). Using this operator ascertains that virtually the whole space of variables is explored. On the other hand, the flipping of a randomly selected bit means a partial loss of genetic information acquired by previous generations. For $m = 0$, the GA performance is strongly limited because there is no means for the given generation to broaden the exploration field (it rather tends to be narrowed due to action of the selection operator). For m exceeding some limit, m_{lim}, the search becomes a 'population-based random-search' method and the search becomes ineffective because of very low speed of action (the value of m_{lim} is indicated as 50% [2]). A compromise m value is needed to ascertain exploration of the whole space with the optimum speed. In algorithms reported in literature, the value of m is either fixed (frequently at ∼1–5% level), or adjusted during calculations. Classical genetic algorithms are, typically, elitist. They exploit three operators: roulette selection, one-point crossover with probability = 1, and mutation.

Search and optimisation methods based on GAs are helpful in solving difficult static and dynamic global tasks in two following categories:

(i) finding a global extremum of a function of a low or moderate number of variables (for example in the static problem of fitting gaussian profiles to multiple peaks in a spectrum);
(ii) finding a solution being better than that which can be found by either deterministic methods or random search (for example a static problem of finding all possible atomic arrangements in very large multicomponent clusters, or a dynamic problem of weather forecast).

The former case concerns the problems where the space size, number of variables, number of local minimums (LMs), calculation time of the fitting-function value are not too large, and constraints are not too complex. The latter case covers the remaining, more difficult tasks.

The genetic approach to global search/optimisation frequently meets the problem of slow convergence (which may limit the real-time applications), of getting stuck in a local minimum, and of low accuracy of the optimum-position determination. Various ways exist which help in improving the GA action:

- using parallel processors;
- hybridising;
- adding new operators;
- modification of the GA through equipping it in some knowledge specific for the given mathematical/physical/engineering or other task.

For an objective function exhibiting many narrow LMs of comparable depth, the search tends to be stopped at one of such minima. For such case, several remedies are known. For example, there exist (related to each other) dynamic penalty function, DPF [5–8], and sequential niching [9] approaches permitting to identify numerous LMs.

Global search and optimisation methods based on the genetic approach find various applications in solving static as well as dynamic computational problems. Examples of surveys and bibliographies of GAs applications can be found, e.g., in refs. [10] (general), [11] (physics and chemistry), [12] (analytical chemistry), [13,14] (engineering), [15] (pattern recognition and machine learning), and [16] (control systems).

Applications of GAs in crystallography started about 10 years ago (see Table 1). Here, the GAs are used, e.g., for molecular design, structure prediction, and solving structures. A large number among the papers listed are devoted to indexing (finding the unit-cell size) and solving the structure (finding the atomic positions and site occupancies) from powder patterns, i.e., to these methods which contribute to determination of crystal structure of polycrystalline materials. Now, for some of the listed crystallographic purposes, commercial GA-based software is available.

Crystallographic studies performed with the use of intense synchrotron radiation sources provide powder-diffraction data of particularly excellent resolution and high counting rate. Solving those coming from large, low-symmetry unit cells frequently requires a considerable computational effort. One can measure a pattern in seconds, minutes or hours, but solving it may require days; some patterns remain unsolved due to limited resolution and computational barriers—that is why the indexing step is sometimes called a bottleneck in structure determination. The development in existing methods [74–78] focuses towards solving patterns of large unit cells, and of specimens including impurity phases. A future progress in indexing techniques is likely to be connected with development of global-optimisation methods.

Indexing is a good GA-efficiency testing object because of complex shape of the objective function: the shape is character-